

```

1 %% Machine Learning
2 % Lab 2: Linear Regression with Multiple variable
3 % — Housing Prices —
4 %{
5 In this part, you will implement linear regression with multiple variables to
6 predict the prices of houses. Suppose you are selling your house and you
7 want to know what a good market price would be. One way to do this is to
8 first collect information on recent houses sold and make a model of housing
9 prices.
10 %}
11
12 %% Initialization
13 clear ; close all; clc % Clear and close figures
14 %% Load Data
15 fprintf('Loading data ...\n');
16 data = load('ex1data2.txt');
17 fprintf('First 5 element of original data:\n')
18 data(1:5, :)
19
20 %% Parsing data
21 X_origin = data(:, 1:2);
22 X = data(:, 1:2); % x1:size (0–2000 feet^2) and x2: number of bedrooms (1–5)
23 Y = data(:, 3); % dollar
24 m = length(Y);
25
26 % Print out some data points
27 fprintf('First 10 examples from the dataset: \n');
28 fprintf(' x = [%0.0f %0.0f], y = %0.0f \n', [X(1:10,:) Y(1:10,:)]);
29
30 %% Scale Features and set them to zero mean
31 % Feature scaling and mean normalization:
32 % Scale features and set them to zero mean
33 fprintf('Normalizing Features ...\n');
34
35 [X avg sigma] = featureNormalize(X);
36
37 % Add intercept term to X
38 X = [ones(m, 1) X];
39
40
41 %% Gradient Descent
42 fprintf('Running gradient descent ...\n');
43
44 % Choose some learning_rate value
45 lr = 0.01; % learning_rate
46 epochs = 400;
47
48 % Init Weights and run Gradient Descent
49 w = zeros(3,1);
50 [w, C_history] = gradientDescentMulti(X, Y, w, lr, epochs);
51
52 %% Plot the convergence graph
53 figure;
54 plot(1:numel(C_history), C_history, '-b', 'LineWidth', 2);
55 % n = numel(A) returns the number of elements, n, in array A, equivalent to prod(
    size(A)).

```

```

56 xlabel('Number of iterations');
57 ylabel('Cost Function (C)');
58
59 % Display gradient descent's result
60 fprintf('Weights computed from gradient descent: \n');
61 fprintf(' %f \n', w);
62 fprintf('\n');
63
64 %% Estimate the price of a 1650 sq-ft, 3 br house
65 % Recall that the first column of X is all-ones. Thus, it does
66 % not need to be normalized.
67 FEET = 1650;
68 BED = 3;
69 price = [1 (FEET-avg(1))/sigma(1) (BED-avg(2))/sigma(2)]*w;
70 %Predicted price should be: $293081.464335
71
72 fprintf(['Predicted price of a 1650 sq-ft, 3 br house ' ...
73         '(using gradient descent):\n $%f\n'], price);

```

```

1 function C = computeCostMulti(X, Y, w)
2 %COMPUTECOSTMULTI Compute cost for linear regression with multiple variables
3
4 % Initialize some useful values
5 m = length(Y);
6 C = 0;
7
8 C=(1/(2*m))*(X*w-Y)'*(X*w-Y);
9 end

```